WHY do we use GAMS even though it is more complex than some other approaches?

1. Algebraic Modeling
   a) Context Changes
   b) Expandability of Models
   c) Adding Conditionals
3. Small to Large Model Development
4. Model Library
5. Use by others
6. Nonlinear and other problem forms
1.a Context Changes

The same algebraic model can be used in multiple contexts often with the same or very similar structure.

- For example, we have two problems:
  - A crop production problem where a farmer wants to maximize the net farm income from growing crops.
  - A manufacturing production problem where a manufacturer wants to maximize the net income from making goods.
1.a Context Changes

SETS Process Production process
/ Corn Corn production
   Wheat Wheat production
   Cotton Cotton production /
Resource Resource item used
/ Land Land used by process
   Labor Labor used by process /
TABLE ResourceUse(Resource,Process) Resource
   Corn Wheat Cotton
   Land  1  1  1 ;
   Labor  6  4  8 ;
PARAMETERS Revenue(Process) Revenues
   / Corn    109
   Wheat    90
   Cotton   115 /
ResourceAvailable(Resource) Resource availability
   / Land    100
   Labor    500 /

VARIABLES
   Profit Net income from process ;
POSITIVE VARIABLES
   Production(Process) Production by crop ;
EQUATIONS
   Objective Maximize farm income
   ResourceEq(Resource) Resource Constraint ;
Objective..
   Profit =E=
   SUM(Process,Revenue(Process)*Production(Process)) ;
ResourceEq(Resource)..
   SUM(Process,ResourceUse(Resource, Process)
   *Production(Process)) =L=
   ResourceAvailable(Resource) ;
MODEL ProfitMax /ALL/;
SOLVE ProfitMax USING LP MAXIMIZING Profit ;
1.b Expandability - Scope

Example

SETS
   Process Production process
   /Corn  Corn production
   Wheat  Wheat production
   Cotton Cotton production
   Soybean Soybeans production /
Resource Resource item used
   /Land  Land used by process
   Labor  Labor used by process /
;

TABLE ResourceUse (Resource, Process) Resource used
   Corn   Wheat   Cotton   Soybean
   Land   1       1       1       1
   Labor  6       4       8       4
;

PARAMETER
   Revenue(Process) Revenues from process production
   /Corn   109
   Wheat   90
   Cotton  115
   Soybean 95 /
   ResourceAvail(Resource) Resource availability
   /Land   100
   Labor  500 /
;

- Instead of growing 3 crops, now a farmer also wants to grow soybeans. One needs only to add an element in
  - set PROCESS
  - table ResourceUse
  - parameter revenues
- Other data and the model structure remains the SAME!
1.b Expandability - Augmenting Existing Models

- In the previous example of crop production, a new constraint is added such as at least 10 units of wheat to be produced for self-consumption.

- Modification includes:
  - adding **data** on minimum land use,
  - and **equation** specification on minimum land use.

**Example**

```
PARAMETER
  Revenue(Process)  Revenues from process production
  /Corn  109
  Wheat  90
  Cotton  115
  Soybean  95 /
  ResourceAvail(Resource)  Resource availability
  /Land  100
  Labor  500 /
  MinLand(Process)  Minimum land requirement
  /Corn  0
  Wheat  10
  Cotton  0
  Soybean  0 /;

EQUATIONS
  Objective   Maximize farm income
  ResourceEq(Resource)  Resource constraint
  MinLandReq(Process)  Minimum land requirement ;

MinlandReq(Process) ..
  ResourceUse("Land", Process) * Production(Process)
  =g= MinLand(Process) ;
```
1.c Adding Conditionals

- Modelers need to be able to write expressions that operate over less than full sets or incorporate various model features conditionally depending on data.
  - Land types (dry land or irrigated land) for crop production
  - Seasonal labor availability
- Such tasks can be accomplished in GAMS using conditionals.
- Several alternatives are available for conditional statements. We will talk about this in detail later if we have time.
2. Self-documenting Nature

- GAMS allows one to add explanatory text when naming SETS, PARAMETERS, TABLES, VARIABLES, EQUATIONS, but it is a good habit to name them with easy-understanding words instead of simple letters.

(A)  
\[
\text{Eq1..} \\
X_1 = \text{SUM}(s_1, y_1(s_1) \times x_2(s_1)) ;
\]

(B)  
\[
\text{Eq2(s2)..} \\
\text{SUM}(s_1, y_3(s_2, s_1) \times x_2(s_1)) = L = y_2(s_2) ; \\
\]

\[
\text{Objective..} \\
\text{Profit} \\
\text{=E=} \\
\text{SUM(\text{Process, Revenue(\text{Process})} \times \text{Production(\text{Process})})} ; \\
\]

\[
\text{Resource Eq(\text{Resource})..} \\
\text{SUM(\text{Process, Resource Use(\text{Resource, Process})} \times \text{Production(\text{Process})})} \\
\text{=L=} \\
\text{Resource Available(\text{Resource})} ; \\
\]
2. Ways to comment your model

- Always remember commenting your code. Comments can not only help others read your code, but also help yourself read for future references.

- Comment a line: asterisk *
- Comment multiple lines: put comments between $ONTEXT and $OFFTEXT
- End of line comment: Use commands $oneolcom and $eolcom //
- In line comment: $inlinecom /& &/
3. Small to large model

- GAMS **expandability** allows the same model structure, calculations, and report writing to be used with SETS with few elements vs. SETS with many items.
- Using a small data set allows up to examine the model structure and function easier and better. Then later one can use same algebra and full problem.

```plaintext
SETS
Stocks POTENTIAL INVESTMENTS /BUYSTOCK1*BUYSTOCK100 /
SmallStocks(Stocks) POTENTIAL INVESTMENTS /BUYSTOCK1*BUYSTOCK2 /
EVENTS EQUALLY LIKELY RETURN SON /EVENT1*EVENT2 /

INVESTAV..
SUM(SmallStocks, PRICES(SmallStocks) * INVEST(SmallStocks))
=L= FUNDS ;
```

- To **expand**, one uses `SmallStocks(Stocks) = YES;`
4. GAMS model library

- GAMS has been used as a standard in optimization models in many fields
- Models exist from experienced users that address similar problems
- Textbook
- Fixing Models Book
- GAMS Library
- GAMS Newsletter
5. Use by others

- GAMS permits one to separate data from the algebraic model, particularly through the use of SAVE, RESTART, and $INCLUDE.
- This feature allows data files to be worked on by other people and also increases run efficiency. One also can use this to separate code functions (e.g. data section, model structure, and report writing).
5. Use by others

- Two files: `data.gms` and `mymodel.gms`.
- Create a `t` folder (or any other names) under the project directory.
- Type the content in the red box before running `data.gms`.
- When finished running `data.gms` which includes all of data, GAMS will save all the information as `~\t\mydata.g00` where it is ready to be used.
5. Use by others

- To solve the model, GAMS retrieves information on data that was saved in ~\t\mydata.g00.
- Type the content in the red box before running mymodel.gms.
- When finished solving mymodel.gms, GAMS will save all information including solutions in ~\t\mymodel.g00 where it is ready to be used later, say, report writing.
Non-linear and Other Problem Forms

- Set declarations
- Parameter declarations
- Variable declarations
- Equation declarations
- Specifying algebraic structure
- Model specifications

Data

Model

Solve specifications

```
SOLVE Ex USING LP MAXIMIZING Z;
SOLVE Ex USING MIP MAXIMIZING Z;
SOLVE Ex USING NLP MAXIMIZING Z;
```
Hands On 3

- Fix errors in Handson3error.gms and send an electronic copy of the correct *.gms file via email.
- Hand in a hard copy of *.lst file (from Solution Report to the end).
Questions?